PORTAL THE ACM DIGITAL LIBRARY

> home > about > feedback > log

US Patent & Trademark Office

Try the *new* Portal design
Give us your opinion after using it.

Search Results

Search Results for: **[((search or query or node or leaf) <near/5> (attribute or field or parameter or key))<AND>(((((hierarch* or tree* or LDAP or "lightweight directory access protocol") <paragraph> (search* or query*)) and (leaf <near/2> (pointer or referenc*) <near/2> (leaf or node)))<AND>(meta_published_date <= 12-01-2000)))]**
Found **29** of **121,350 searched.**

Search within Results

> Advanced Search > Search Help/Tips

Sort by: Title Publication Publication Date Score Binder

Results 1 - 20 of 29 short listing

Prev Page 1 2 Next Page

1 The hB-tree: a multiattribute indexing method with good guaranteed performance 100%

David B. Lomet , Betty Salzberg

ACM Transactions on Database Systems (TODS) December 1990

Volume 15 Issue 4

A new multiattribute index structure called the hB-tree is introduced. It is derived from the K-D-B-tree of Robinson [15] but has additional desirable properties. The hB-tree internode search and growth processes are precisely analogous to the corresponding processes in B-trees [1]. The intranode processes are unique. A k-d tree is used as the structure within nodes for very efficient searching. Node splitting

requires that this k-d tree be split. This produces nodes which no longer represe ...

2 Access methods for multiversion data 100%

David Lomet , Betty Salzberg

ACM SIGMOD Record , Proceedings of the 1989 ACM SIGMOD international conference on Management of data June 1989

Volume 18 Issue 2

We present an access method designed to provide a single integrated index structure for a versioned timestamped database with a non-deletion policy. Historical data (superceded versions) is stored separately from current data. Our access method is called the Time-Split B-tree. It is an index structure based on Malcolm Easton's Write Once B-tree. The Write Once B-tree was developed for data stored entirely on a Write-Once Read-Many or WORM optical ...

3 A Cost Model for the Internal Organization of B+-Tree Nodes 100%

Wilfred J. Hansen

ACM Transactions on Programming Languages and Systems (TOPLAS) October 1981

Volume 3 Issue 4

4 Ubiquitous B-Tree 100%

Douglas Comer

ACM Computing Surveys (CSUR) June 1979

Volume 11 Issue 2

5 The DataIndex: a structure for smaller, faster data warehouses 100%

Anindya Datta , Igor Viguier

ACM SIGMIS Database September 1998

Volume 29 Issue 4

In this paper we present DataIndexes, a family of design strategies for data warehouses to support OnLine Analytical Processing (OLAP). As the name implies, DataIndexes are both a storage structure for the warehoused relational data and an indexing scheme to provide fast

access to that data. We present two simple DataIndexes: the Basic DataIndex (BDI), which can be used for any attribute and the Join DataIndex (JDI), which is used for foreign-key attributes. Either structure can be shown to sign ...

6 Distance browsing in spatial databases 100%

Gísli R. Hjaltason , Hanan Samet

ACM Transactions on Database Systems (TODS) June 1999 Volume 24 Issue 2

We compare two different techniques for browsing through a collection of spatial objects stored in an R-tree spatial data structure on the basis of their distances from an arbitrary spatial query object. The conventional approach is one that makes use of a k-nearest neighbor algorithm where k is known prior to the invocation of the algorithm. Thus if m < k neighbors are needed, the k-nearest neighbor alg ...

7 A symmetric concurrent B-tree algorithm 100%

Vladimir Lanin , Dennis Shasha

Proceedings of 1986 fall joint computer conference on Fall joint computer conference November 1999

8 Multidimensional access methods 100%

Volker Gaede , Oliver Günther

ACM Computing Surveys (CSUR) June 1998 Volume 30 Issue 2

Search operations in databases require special support at the physical level. This is true for conventional databases as well as spatial databases, where typical search operations include the point query (find all objects that contain a given search point) and the region query (find all objects that overlap a given search region). More than ten years of spatial database research have resulted in a great variety of multidimensional access methods to support ...

9 Storing a collection of polygons using quadtrees 100%

Hanan Samet , Robert E. Webber

ACM Transactions on Graphics (TOG) July 1985
Volume 4 Issue 3
An adaptation of the quadtree data structure that represents polygonal maps (i.e., collections of polygons, possibly containing holes) is described ina manner that is also useful for the manipulation of arbitrary collections of straight line segments. The gol is to store these maps without the loss of information that results from digitization, and to obtain a worst-case execution time that is not overly sensitive to the positioning of the map. A regular decomposition variant of the region ...

10 Implementing deletion in B+-trees 100%
Jan Jannink
ACM SIGMOD Record March 1995
Volume 24 Issue 1
This paper describes algorithms for key deletion in B^+-trees. There are published algorithms and pseudocode for searching and inserting keys, but deletion, due to its greater complexity and perceived lesser importance, is glossed over completely or left as an exercise to the reader. To remedy this situation, we provide a well documented flowchart, algorithm, and pseudo-code for deletion, their relation to search and insertion algorithms, and a reference to a freely available, complete ...

11 An efficient directory system for document retrieval 100%
D. Motzkin
Proceedings of the 14th annual international ACM SIGIR conference on Research and development in information retrieval September 1991

12 A robust B-tree implementation 99%
J. P. Black , D. J. Taylor , D. E. Morgan
Proceedings of the 5th international conference on Software engineering March 1981
A storage structure for B-trees is presented which is robust in that any pair of changes to structural fields of an instance of the structure

can be detected, as well as many sets of larger numbers of changes. Included in the paper are a motivation for robustness as a design criterion, cost and performance implications of the B-tree implementation, and a solution to the subproblem of a robust implementation for each node's contiguous list of pointers and keys.

13 The hB $^\Pi$-tree: a multi-attribute index supporting concurrency, recovery and node consolidation 99%

Georgios Evangelidis , David Lomet , Betty Salzberg

The VLDB Journal — The International Journal on Very Large Data Bases February 1997

Volume 6 Issue 1

We propose a new multi-attribute index. Our approach combines the hB-tree, a multi-attribute index, and the Π-tree, an abstract index which offers efficient concurrency and recovery methods. We call the resulting method the hB $^\Pi$-tree. We describe several versions of the hB $^\Pi$-tree, each using a different node-splitting and index-term-posting algorithm. We also describe a new node deletion algorithm. We have implemented all the versions of the hB $^\Pi$-tree. Our performance results ...

14 Implementing a generalized access path structure for a relational database system 99%

Theo Haerder

ACM Transactions on Database Systems (TODS) September 1978

Volume 3 Issue 3

A new kind of implementation technique for access paths connecting sets of tuples qualified by attribute values is described. It combines the advantages of pointer chain and multilevel index implementation techniques. Compared to these structures the generalized access path structure is at least competitive in performing retrieval and update operations, while a considerable storage space saving is gained. Some additional features of this structure support m-way joins and th ...

15 The PSG system: from formal language definitions to interactive programming environments 99%

Rolf Bahlke , Gregor Snelting
ACM Transactions on Programming Languages and Systems (TOPLAS) August 1986
Volume 8 Issue 4
The PSG programming system generator developed at the Technical University of Darmstadt produces interactive, language-specific programming environments from formal language definitions. All language-dependent parts of the environment are generated from an entirely nonprocedural specification of the language's syntax, context conditions, and dynamic semantics. The generated environment consists of a language-based editor, supporting systematic program development by named program fragments, ...

16 H-trees: a dynamic associative search index for OODB 99%
Chee Chin Low , Beng Chin Ooi , Hongjun Lu
ACM SIGMOD Record , Proceedings of the 1992 ACM SIGMOD international conference on Management of data June 1992
Volume 21 Issue 2
The support of the superclass-subclass concept in object-oriented databases (OODB) makes an instance of a subclass also an instance of its superclass. As a result, the access scope of a query against a class in general includes the access scope of all its subclasses, unless specified otherwise. To support the superclass-subclass relationship efficiently, the index must achieve two objectives. First, the index must support efficient retrieval of instances from a single class. Second, it must ...

17 Extendible hashing—a fast access method for dynamic files 99%
Ronald Fagin , Jurg Nievergelt , Nicholas Pippenger , H. Raymond Strong
ACM Transactions on Database Systems (TODS) September 1979
Volume 4 Issue 3
Extendible hashing is a new access technique, in which the user is guaranteed no more than two page faults to locate the data associated with a given unique identifier, or key. Unlike conventional hashing,

extendible hashing has a dynamic structure that grows and shrinks gracefully as the database grows and shrinks. This approach simultaneously solves the problem of making hash tables that are extendible and of making radix search trees that are balanced. We study, by analysis and simulatio ...

18 Finger search trees with constant insertion time 99%
Gerth Stølting Brodal
Proceedings of the ninth annual ACM-SIAM symposium on Discrete algorithms January 1998

19 Fast address lookups using controlled prefix expansion 99%
V. Srinivasan , G. Varghese
ACM Transactions on Computer Systems (TOCS) February 1999

Volume 17 Issue 1
Internet (IP) address lookup is a major bottleneck in high-performance routers. IP address lookup is challenging because it requires a longest matching prefix lookup. It is compounded by increasing routing table sizes, increased traffic, higher-speed links, and the migration to 128-bit IPv6 addresses. We describe how IP lookups and updates can be made faster using a set of of transformation techniques. Our main technique, controlled prefix expansion, transf ...

20 Faster IP lookups using controlled prefix expansion 99%
V. Srinivasan , George Varghese
ACM SIGMETRICS Performance Evaluation Review , Proceedings of the 1998 ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems June 1998
Volume 26 Issue 1
Internet (IP) address lookup is a major bottleneck in high performance routers. IP address lookup is challenging because it requires *a longest matching prefix* lookup. It is compounded by increasing routing table sizes, increased traffic, higher speed links, and the migration to 128 bit IPv6 addresses. We describe how IP

lookups can be made faster using a new technique called *controlled prefix expansion*. Controlled prefix expansion, together with optimization techniques based on dyn ...

Results 1 - 20 of 29 short listing

Prev Page 1 2 Next Page

The ACM Portal is published by the Association for Computing Machinery.
Copyright © 2003 ACM, Inc.

Try the *new* Portal design
Give us your opinion after using it.

Search Results

Nothing Found

Your search for **[((LDAP or "lightweight directory access protocol" or (network <near/2> directory)) and (leaf <near/3> (pointer or referenc*) <near/3> node))<AND>(meta_published_date <= 12-01-2000)]** did not return any results.

You may revise it and try your search again below or click advanced search for more options.

```
((LDAP or "lightweight directory
access protocol" or (network
<near/2> directory)) and (leaf
<near/3> (pointer or referenc*)
<near/3>
node))<AND>(meta_published_date <=
12-01-2000 )
```

SEARCH [**Advanced Search**] [**Search Help/Tips**]

Complete Search Help and Tips

The following characters have specialized meaning:

Special Characters	Description
, () [	These characters end a text token.
= > < !	These characters end a text token because they signify the start of a field operator. (! is special: != ends a token.)
` @ \Q < { [!	These characters signify the start of a delimited token. These are terminated by the end character associated with the start character.

Search Results for: **[((LDAP or "lightweight directory access protocol" or (network <near/2> directory)) and ((leaf or node) <near/3> (pointer or referenc*)))<AND>(meta_published_date <= 12-01-2000)]**
Found **6** of **121,350 searched.**

Search within Results

1 Tempest and typhoon: user-level shared memory 100%
S. K. Reinhardt , J. R. Larus , D. A. Wood
ACM SIGARCH Computer Architecture News , Proceedings of the 21ST annual international symposium on Computer architecture April 1994
Volume 22 Issue 2
Future parallel computers must efficiently execute not only hand-coded applications but also programs written in high-level, parallel programming languages. Today's machines limit these programs to a single communication paradigm, either message-passing or shared-memory, which results in uneven performance. This paper addresses this problem by defining an interface, *Tempest*, that exposes low-level communication and memory-system mechanisms so programmers and compilers can customize polici ...

2 Mechanisms for cooperative shared memory 99%

David A. Wood , Satish Chandra , Babak Falsafi , Mark D. Hill , James R. Larus , Alvin R. Lebeck , James C. Lewis , Shubhendu S. Mukherjee , Subbarao Palacharla , Steven K. Reinhardt

ACM SIGARCH Computer Architecture News , Proceedings of the 20th annual international symposium on Computer architecture May 1993

Volume 21 Issue 2

We introduce a new organization for multi-bank caches: the skewed-associative cache. A two-way skewed-associative cache has the same hardware complexity as a two-way set-associative cache, yet simulations show that it typically exhibits the same hit ratio as a four-way set associative cache with the same size. Then skewed-associative caches must be preferred to set-associative caches. Until the three last years external caches were used and their size could be relatively large. P ...

3 On effective multi-dimensional indexing for strings 99%

H. V. Jagadish , Nick Koudas , Divesh Srivastava

ACM SIGMOD Record , Proceedings of the 2000 ACM SIGMOD international conference on Management of data May 2000

Volume 29 Issue 2

As databases have expanded in scope from storing purely business data to include XML documents, product catalogs, e-mail messages, and directory data, it has become increasingly important to search databases based on wild-card string matching: prefix matching, for example, is more common (and useful) than exact matching, for such data. In many cases, matches need to be on multiple attributes/dimensions, with correlations between the dimensions. Traditional multi-dimensional index structures, ...

4 Where is time spent in message-passing and shared-memory programs? 99%

Satish Chandra , James R. Larus , Anne Rogers

Proceedings of the sixth international conference on Architectural support for programming languages and operating systems November 1994
Volume 29 , 28 Issue 11 , 5
Message passing and shared memory are two techniques parallel programs use for coordination and communication. This paper studies the strengths and weaknesses of these two mechanisms by comparing equivalent, well-written message-passing and shared-memory programs running on similar hardware. To ensure that our measurements are comparable, we produced two carefully tuned versions of each program and measured them on closely-related simulators of a message-passing and a shared-memory machine, ...

5 An empirical comparison of the Kendall Square Research KSR-1 and Stanford DASH multiprocessors 99%
J. P. Singh , T. Joe , J. L. Hennessy , A. Gupta
Proceedings of the 1993 ACM/IEEE conference on Supercomputing December 1993

6 Partitioning in X.500 99%
John Henshaw , Michael Bauer
Proceedings of the 1993 ACM/SIGAPP symposium on Applied computing: states of the art and practice March 1993

Results 1 - 6 of 6 short listing

The ACM Portal is published by the Association for Computing Machinery.
Copyright © 2003 ACM, Inc.